IN THE CLAIMS

1.   (Previously amended) A method of generating code for Enterprise JavaBean (EJB) components from a business process, comprising the steps of:

graphically modeling said business process using a UML drawing tool to provide an UML model having a plurality of EJB Classes;

defining relationships between said plurality of EJB classes;

stereotyping each of said plurality of EJB classes into one or more EJB components;

transforming each of said EJB components into EJB source code; and

embedding code markers in said EJB source code to enable subsequent updates to said EJB source code.

2.   (Previously presented)   The method of claim 1, further comprising the step of compiling said EJB source code to generate EJB application in accordance with deployment properties.

3.   (Previously presented)   The method of claim 2, further comprising the step of deploying said EJB application to a server using one of the following: bean managed persistence or container managed persistence.

4.   (Previously amended) The method of claim 1, wherein the step of stereotyping stereotypes an EJB class into at least one of the following Smart EJB component: Belonging, Session, Entity, Configurable Entity, Business Policy and Workflow.

5.   (Previously presented)   The method of claim 4, wherein an Entity EJB component comprises at least one interface and two EJB classes.

6.   (Previously presented)   The method of claim 5, wherein said Entity EJB component being associated with a Primary Key class and a Value class.

7. (Previously presented) The method of claim 1, wherein each EJB component includes at least one of the following: name, stereotype, attribute and method.

8. (Previously presented) The method of claim 7, wherein each attribute includes a pair of accessor methods.

9. (Previously presented) The method of claim 1, wherein said relationships includes at least one of the following: inheritance and aggregation.

10. (Previously presented) The method of claim 9, wherein said aggregation includes multiplicity.

11. (Previously presented) The method of claim 10, further comprising the steps of:
determining if said multiplicity relationship is one to many; and
stereotyping said aggregation relationship into a collection type if it is determined that said multiplicity relationship is one to many.

12. (Previously presented) The method of claim 11, wherein said collection type includes one of the following: Set, Array, List or Map.

13. (Previously presented) The method of claim 1, wherein each EJB component is a Smart Component having at least one Smart Feature.

14. (Previously presented) The method of claim 13, wherein said Smart Feature includes one of the following: SmartKey, SmartHandle and SmartValue.

15. (Previously presented) The method of claim 1, wherein said Smart component is an eBusiness Smart Component.

16. (Previously presented) The method of claim 1, wherein the step of transforming includes the step generating said EJB codes according to a Code Template Dictionary.

17. (Previously presented) The method of claim 16, wherein said Code Template Dictionary includes key-value pair entries.

25584815_1.DOC

3

18.   (Previously presented)   The method of claim 17, wherein values of said Code Template Dictionary represent EJB code templates.

19.   (Previously presented)   The method of claim 1, wherein the step of embedding includes the step of adding business logic code between said code markers.

20.   (Currently amended) The method of claim 19, further comprising the step of synchronizing said UML model with said business logic code, thereby providing ~~round trip engineering~~ support <u>for iterative development cycle</u>.

25584815_1.DOC

4